REMOTE HAZE MONITORING BY SATELLITE

A study to demonstrate the feasibility and
determine the optimum method

Ernest H. Rogers
Space Physics Laboratory
The Aerospace Corporation
P. O. Box 92957
Los Angeles, California 90009

March 1973
Interim Report for Period May - September 1972

Prepared for
Goddard Space Flight Center
Greenbelt, Maryland 20771

PREFACE

Progress during the period 10 May - 30 September 1972 in the Study to Demonstrate the Feasibility of and Determine the Optimum Method of Remote Haze Monitoring by Satellite is described in this report. Accomplishments in this period included preparation of computer software, establishment of ground truth data collection routines, design and construction of two solar aureole monitors, visual and photographic data collection, and operation of the above observation methods during several ERTS-1 passes. No problems were encountered that will impede the progress of the study.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

PROGRESS REPORT

INTRODUCTION

Progress during the period 10 May - 30 September 1972 in the Study to Demonstrate the Feasibility of and Determine the Optimum Method of Remote Haze Monitoring by Satellite is described in this report. Accomplishments in this period included preparation of computer software, establishment of ground truth data collection routines, design and construction of two solar aureole monitors, visual and photographic data collection, and operation of the above observation methods during several ERTS-1 passes. No problems were encountered that will impede the progress of the study.

Computer Software

The required computer software was written and debugged. This includes a subroutine to convert the IBM 360-370 format to that of the CDC 7600, a general analysis program, and several specific analysis subroutines. A sample tape, received from GSFC, was processed successfully.

The computer program prepared for the analysis of ERTS MSS data is quite flexible. This program has been written with a view towards incorporating different and sophisticated analysis procedures as simply as possible by changing only a few subroutines. The first analysis procedure was simply a printout of the intensity and a gray scale plot of this intensity. The present analysis procedure provides gray scale plots (pictures) of selected strips from the MSS frame and provides a line plot of the intensity for all four bands for selected adjacent lines within the strip.

The flow of operations in the basic program is shown in Figure 1. The names of the subroutines performing the various operations are indicated under the lower right corner of the box containing the operations descriptions.

The spatial region to be analyzed is selected by cards read by the routine DESLOC. At present this is specified by giving the beginning and ending scan line numbers in the frame and the beginning and ending point numbers in the scan line. This routine also recognizes a stop directive on the input cards.

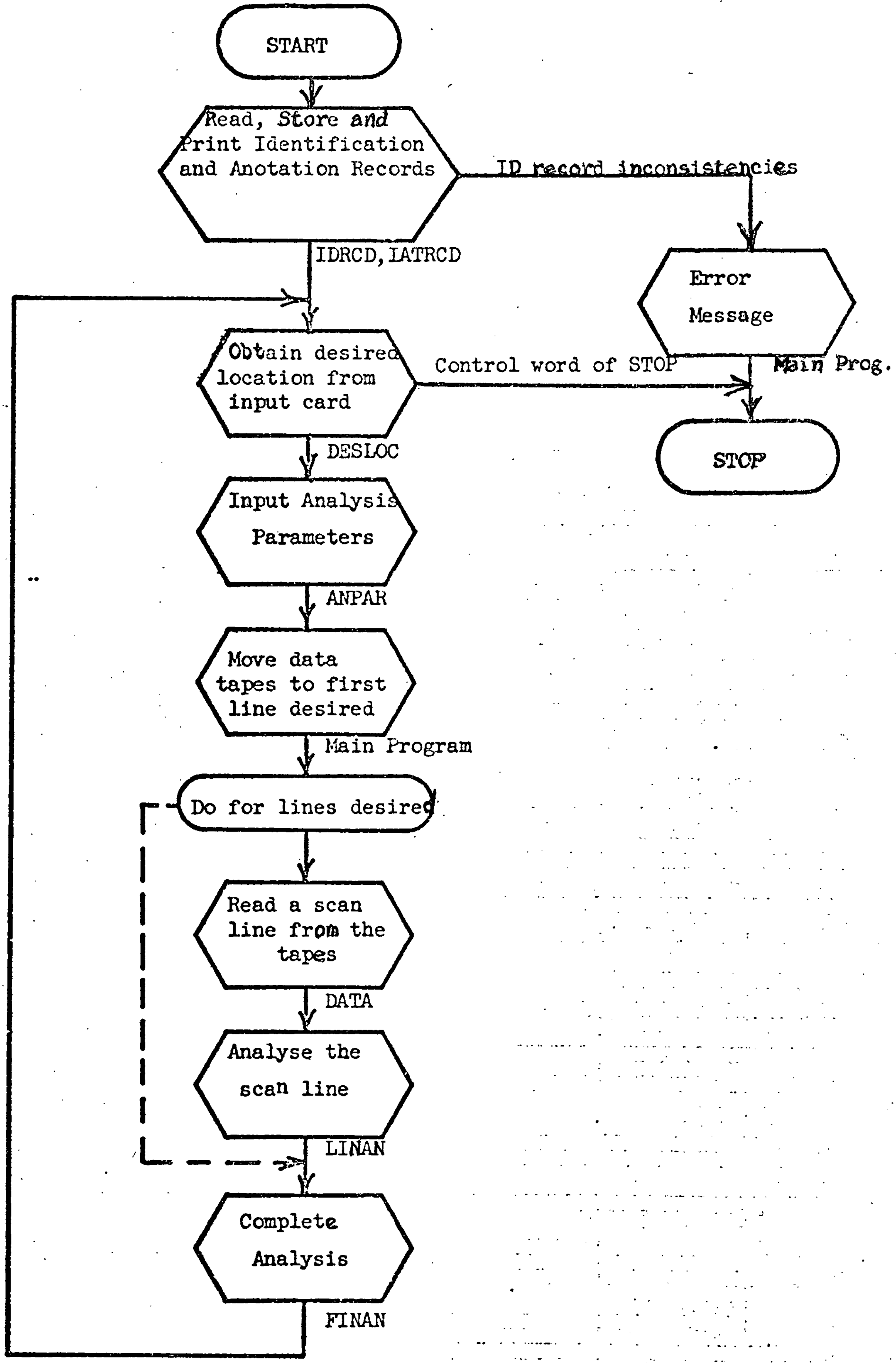
START
Read, Store and Print Identification and Anotation Records
ID record inconsistencies
IDRCD, IATRCD
Error Message
Main Prog.
STOP
Obtain desired location from input card
Control word of STOP
DESLOC
Input Analysis Parameters
ANPAR
Move data tapes to first line desired
Main Program
Do for lines desired
Read a scan line from the tapes
DATA
Analyse the scan line
LINAN
Complete Analysis
FINAN

Figure 1. Flow chart of the ERTS1 program.

The analysis part of the program consists of 3 subroutines (or 3 entry points in one subroutine). The names of the entry points are ANPAR, LINAN and FINAN. The present analysis processes entire lines and ignores any point information which may be provided from DESLOC. For all the lines indicated to DESLOC a gray scale plot is provided for one spectral band. For a selected group of lines within this strip the intensity is plotted for all 4 spectral bands on a line plot as a function of point number. The entry point ANPAR is called before the analysis begins and permits the input of various parameters needed for the analysis. In the present version the parameters required are the beginning and ending lines for which line plots are desired and the spectral band number which is to be gray scale plotted. The program then begins to deliver, one scan line at a time, the MSS data for all four bands for the entire scan line to the entry point LINAN. The present program saves the information for a gray scale plot if desired, and then determines if this is a scan line for which a line plot is desired. If so the plot is prepared for the entire line. After all the desired lines have been analyzed, the entry point FINAN is called to do any summarizing needed to complete the analysis procedure. In the present program this consists of generating the gray scale plot.

DESLOC is called again after the call to FINAN. An attempt is made to input another spatial location and the whole procedure is repeated. Since the data is on tapes in sequential scan line order it is much more efficient to order the locations desired for analysis in ascending order of line number. The program will handle out of order requests correctly but at the expense of rewinding files and skiping records which would otherwise not be necessary.

All communication between subroutines is by means of common blocks. The identification (ID) and annotation (IAT) records are in labelled common blocks /IDRC/ and /IATBLK/ respectively. A common block called /CAL/ contains the calibration information from the data records. The scan line and data point numbers are transferred in a common block /LPOS/. The data from the scan line is stored in a blank common array.

Input of desired location and band number are by means of a free form card technique. The words and variables recognized are shown in Table 1. Some examples may clarify the use. Each line in the following example corresponds to one card on the input file. Spaces and column locations are unimportant. The punctuation shown is required. No spaces may exist in words or numbers. The order with which parameters are sepcified on a single card is unimportant.

Table 1. Words and Parameters Recognized by the Initial Version of the ERTS Data Analysis Program

Routine Name	Control Words	Parameters		Function
DESLOC	STOP	None		Stops the program
	LINE	LBEG = nnn	The beginning scan line number	Inputs desired spatia location
		LEND = nnn	The ending scan line number	
		LDEL = nnn	The number of scan lines desired	Only one of these need be specified
		JBEG = nnn	The beginning point number in the scan line	
		JEND = nnn	The ending point number in the scan line	Only one of these need be specified
		JDEL = nnn	The number of points desired from a scan line	
ANPAR	None	IBAND = nnn	The MSS band number desired for gray scale plotting	
		PBEG = nnn	The beginning line number for which a line plot of the intensity in all four bands is desired	

Routine Name	Control Words	Parameters	Function
		PEND = nnn	The ending line number for which a line plot of the intensity in all four bands is desired

Example:

```
LINE (LBEG = 3, LEND = 25, JBEG = 1, JEND = 900)
     (IBAND = 4, PBEG = 7, PEND = 11)
LINE (LBEG = 400, LEND = 450)
     (PBEG = 405, PEND = 410)
LINE (LBEG = 500, LDEL = 750)
     (IBAND = 1)
STOP
```

This sequence of data cards will, with the present program, make a gray scale plot of spectral 4 for scan lines 3 to 25. Intensity line plots will be prepared for lines 7 through 11. The location of these intensity lines on the gray scale plot will be indicated by outlining them with a constant intensity line. Then intensity line plots will be prepared for scan lines 405 through 410. No gray scale plot will be generated. Finally, a gray scale plot of spectral band one for scan lines 500 to 750 will be prepared.

The difference in the computer word format between the XDS computers at the ERTS NDPF and the CDC computers available at Aerospace made the writing of these programs somewhat more difficult than it might have been otherwise. Since the XDS format is essentially the IBM 360-370 format it was deemed desirable to create a somewhat more general purpose routine for the conversion of data from XDS format to CDC format, than might have been required strictly for the ERTS analysis. This routine is described in the appendix. Copies of the programs are available from the investigators.

Ground Truth

Four types of ground truth data are obtained:

a) Visibility data from airports

b) Air pollution control district data

c) Aureole monitor

d) Visual and photographic data

Altitude of the top of the haze layer and vertical.temp profiles

The first two are obtained from outside ongoing sources. The last two are actively gathered by the contractor specifically for this analysis.

Visibility Data from Airports

Airport visibility is determined hourly on the hour, and broadcast continuously for the use of private and commercial pilots.

ERTS-1 passes over our test site between 5 minutes before and one minute after the hour. Thus the time of visibility observations coincides nicely with ERTS-1 passes.

Visibility reports are recorded at the time of every ERTS-1 pass over our test area for the following airports, the locations of which are shown in Figure 2.

1	Los Angeles International
2	Burbank
3	Van Nuys
4	Ontario
5	Orange County
6	Long Beach
7	Torrance
8	Palmdale
9	Palm Springs

This visibility data is not expected to be uniquely correlated with the effects of haze on the ERTS data, because first it is a measure of horizontal visibility and does not reflect the vertical profile of the haze layer, secondly it is a semi-subjective measurement. It is included in our package of ground truth because it is expected to be qualitatively correlated with total haze. Also, because it is readily available from a large number of sites, one wonders if this data might be usefull in making first order corrections and remove the effects of haze on ERTS imagery. Therefore it will be interesting to correlate these visibility data with the effects of haze on ERTS data.

Air Pollution Control District Data

The Air Pollution Control District, County of Los Angeles, records the amounts of various pollutants at ground level at 12 stations hourly. The locations of these stations, designated by letters A-L, are shown in Figure 2.

The pollutants measured are:

NO (pphm)
O_3 (pphm)
Hydrocarbons (ppm)
CO (ppm)
Methane (ppm)
SO_2 (pphm)
Particulates (KM units x 10)
NO_2 (pphm)

Much of the haze over the Los Angeles area is composed of smog. Therefore, it will be interesting to see the extent to which these various pollutants are correlated with the effects of haze on the ERTS data.

Solar Aureole Monitor

An aureole monitor has been designed, and two units have been built. This instrument measures the brightness of the solar aureole at angular distances of 1-1/2, 3, 6 degrees from the center of the sun in the almucantar, i.e., parallel to the horizon. The spectral response extends from 0.6-1.0 microns,

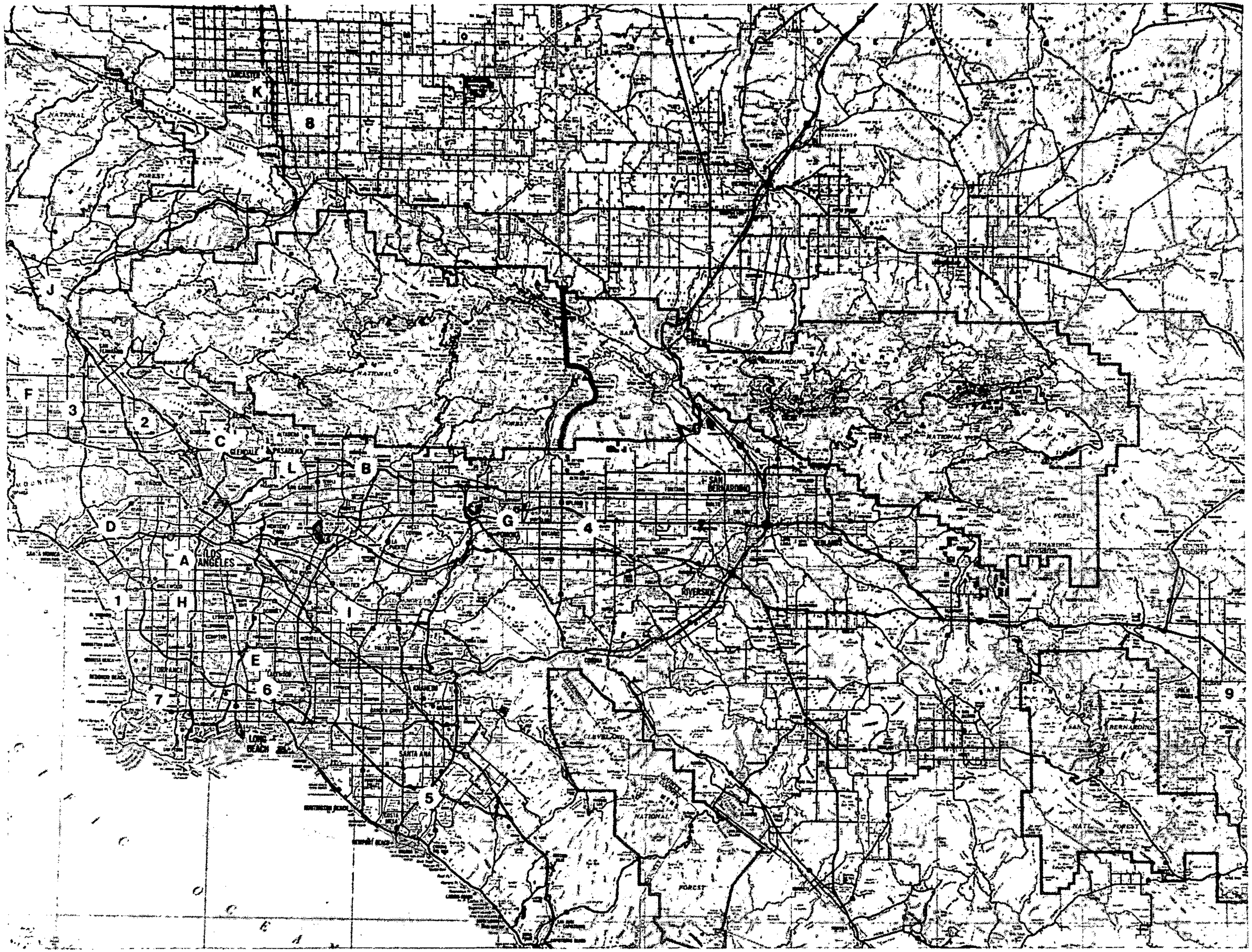

Fig. Map of the Test Area showing airports where visibility data is obtained (1-9), and the locations where pollution is monitored (A-L).

3

The solar aureole results from small angle scattering of sunlight by aerosols, thus it is an excellent monitor of the total integrated number of haze producing aerosols along the path between the observer and the sun.[1,2] This same total integral determines the effect of haze on ERTS data. Measurements of the size distribution of the aerosols can, in principle, be obtained by measurement of the solar aurole in several well defined spectral intervals. Such an analysis is beyond the scope of this work. The spectral interval was chosen to be nearly the same as the sum of ERTS-1 sensors in order to maximize the correlation of these measurements with the effect of haze on ERTS data. The green spectral region was eliminated in order to reduce the effects of Rayleigh scattering in the aureole measurements.

Figure 3 is a sketch of the aureole monitor. The incoming sunlight passes through the entrance aperture, A. A pinhole image of the solar disc passes through the solar disc aperture, O, into the light trap. Pinhole images of the aureole are detected at angles of 1-1/2, 3, 6 degrees from the center of the solar disc by the detectors. These detectors are 1 mm diameter silicon diodes with a #29 filter mounted on the face of each diode.

Typical aureole intensities are in the range of 10^{-4} - 10^{6} of the brightness of the solar disc. Thus, the primary technical difficulty in constructing an aureole monitor is insuring that a negligible amount of the large flux of the solar disc scatters or diffracts into the aureole detectors. A pinhole was chosen for the entrance aperture rather than a lens, because a lens at that position would need to be inherently a very low scatter fabrication and would need to be kept extremely clean. Diffraction of light from the pinhole into the aureole detectors can be calculated, and thus designed to be negligibly small. The ratio of diffraction to incident light is given by:[3]

$$I/I_o = \frac{2Ji(x)}{(x)} \qquad (1)$$

where

$$X = \frac{\pi D \operatorname{Sin}\theta}{\lambda}$$

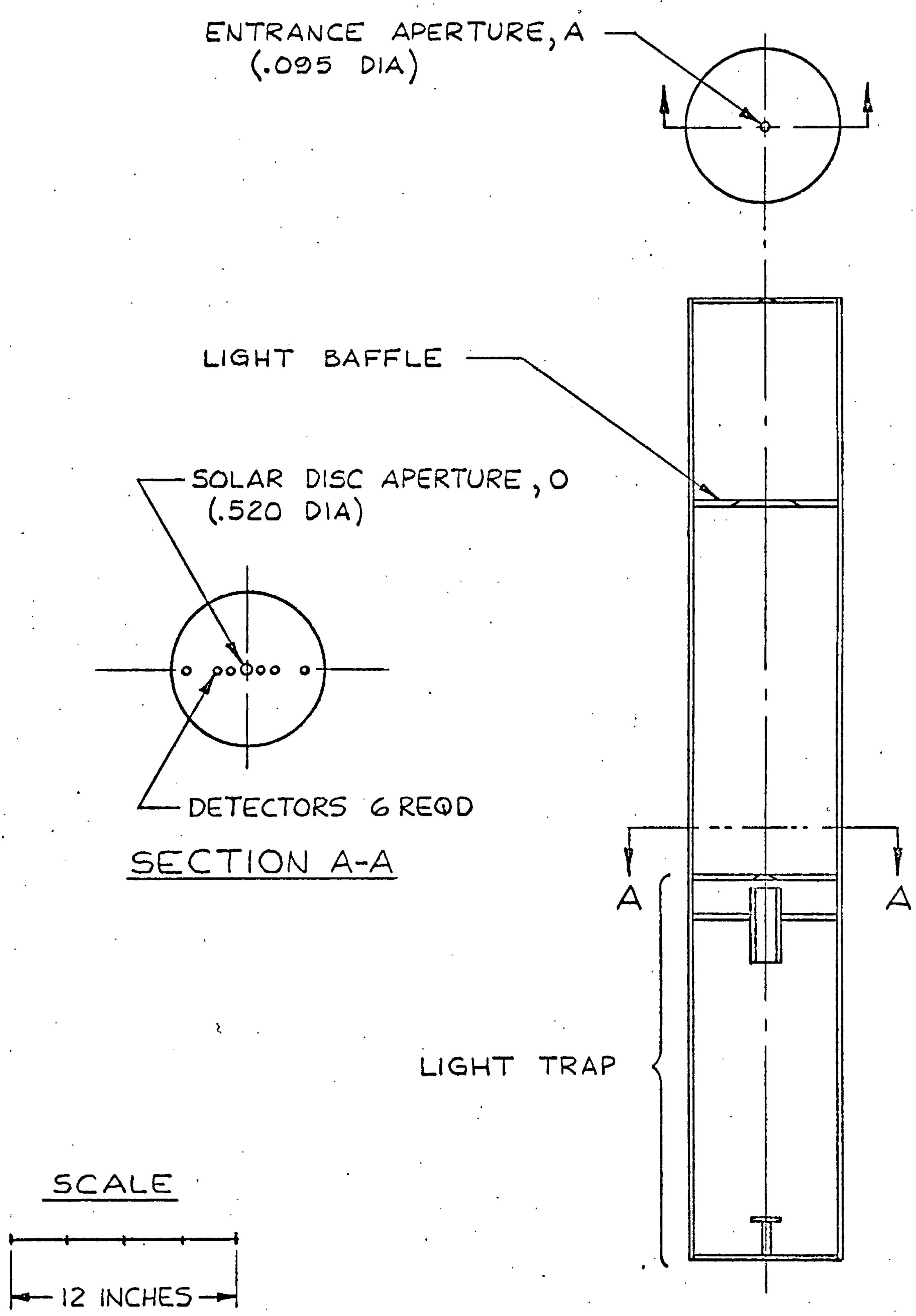

Fig. 3. Scale drawing of the essential elements of the solar aureole monitor.

Ji is the first order Bessel function

D is the diameter of the aperture

θ is the angle of diffraction

λ is the wavelength of light

Eq. (1) has numerous maxima and minima. Its max value falls below 10^{-6} for $X \approx 150$, and continues to fall as X increases.

For $D = 0.095'' = 2.413 \times 10^{3}\mu$

$\lambda = 0.75\mu$

$X = 150$

$\theta = 51$ deg

The solar disc subtends a half angle of 16 degrees. This is smeared by the finite size of the entrance aperture to 21 degrees at the detector plane. Thus light diffraction into the 1-1/2 degree detector is much less than 10^{-6} even from the edge of the solar image next to that detector. The light diffracted into the 3 and 6 degree detectors is much less even then that scattered into the 1-1/2 degree detector. Therefore, diffraction from the entrance aperture is negligible compared with the aureole signal in this system.

The solar disc aperture, O, allows the intense flux of the solar image to pass into the light trap. The diameter of the solar disc at this point is 0.28". This is blurred to a diameter of 0.375" by the finite size of the entrance aperture. It is smeared even farther to about 0.40" by diffraction from the entrance aperture. The diameter of the aperture O of 0.52" allows some margin. A separate unit, boresighted with the aureole monitor, is used to keep the solar image centered in the solar disc aperture. This also maintains the aureole detectors at the proper angles. The design of the light trap baffles etc., limits the light from the solar disc that scatters inside the instrument into the aureole detectors to $\approx 2 \times 10^{-10}$ of the intensity of the image of the solar disc.

This is felt to be an appropriate quantitative monitor of haze producing aerosols. In practice it is limited by the fact that only 2 predetermined locations can be covered during each ERTS pass, whereas our test area covers many areas of vastly different haze conditions. Figure 4 shows an aureole monitor in operation.

Visual and Photographic Data. Altitude of the top of the haze layer and verticle temperature profiles.

Flights are made over the test area, at the times of ERTS passes, in a light plane, Cessna 172 or 150. The altitude is typically 6,000 ft, but in any case above the haze layer.

Visual observations of the location, extent and relative amounts of haze are recorded.

Photographs are taken to document the visual observations and to aid in establishing relative amounts of haze from pass to pass. These photographs are taken using a 35 mm camera with a 50 mm lens and a UV filter. Kodachrome II film, processed by Kodak, is used. Photographs are taken looking both down, and towards the horizon in a variety of directions. Time, location, altitude, direction, and camera settings are noted.

A large part of the test area can be observed within 30 minutes before and 30 minutes after the ERTS pass in this manner.

Even though this method is subjective, the investigators feel that it is the best available method for establishing the distribution of haze over the test area at the time of an ERTS pass.

At selected spots, the altitude of the top of the haze layer is determined by flying along the top of the layer and recording the altimeter reading. The airport visibility data is expected to be much more useful when combined with this haze thickness measurement. Vertical temperature profiles are taken at some spots by recording aircraft external temperature vs altitude.

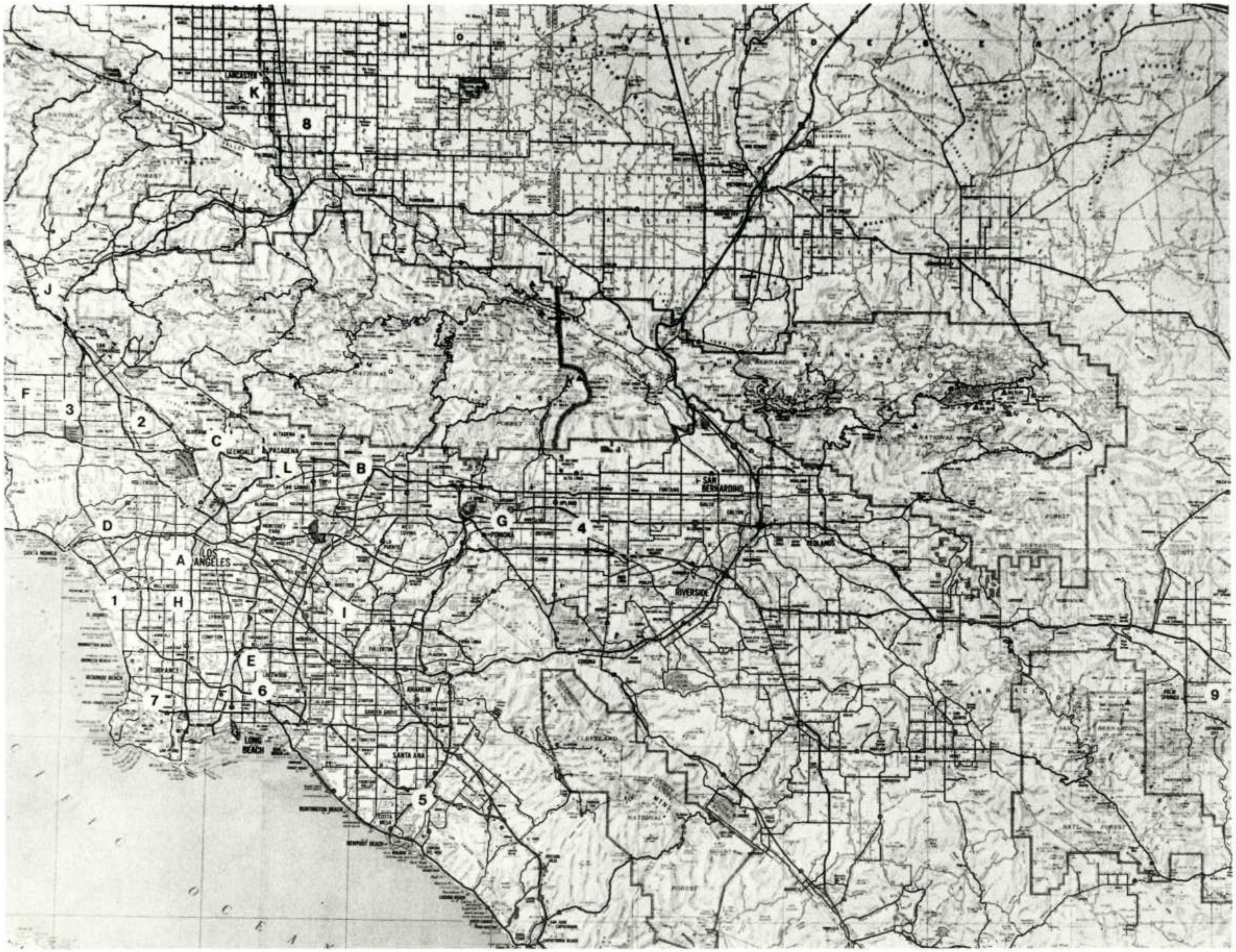

Fig. Map of the Test Area showing airports where visibility data is obtained (1-9), and the locations where pollution is monitored (A-L).

B

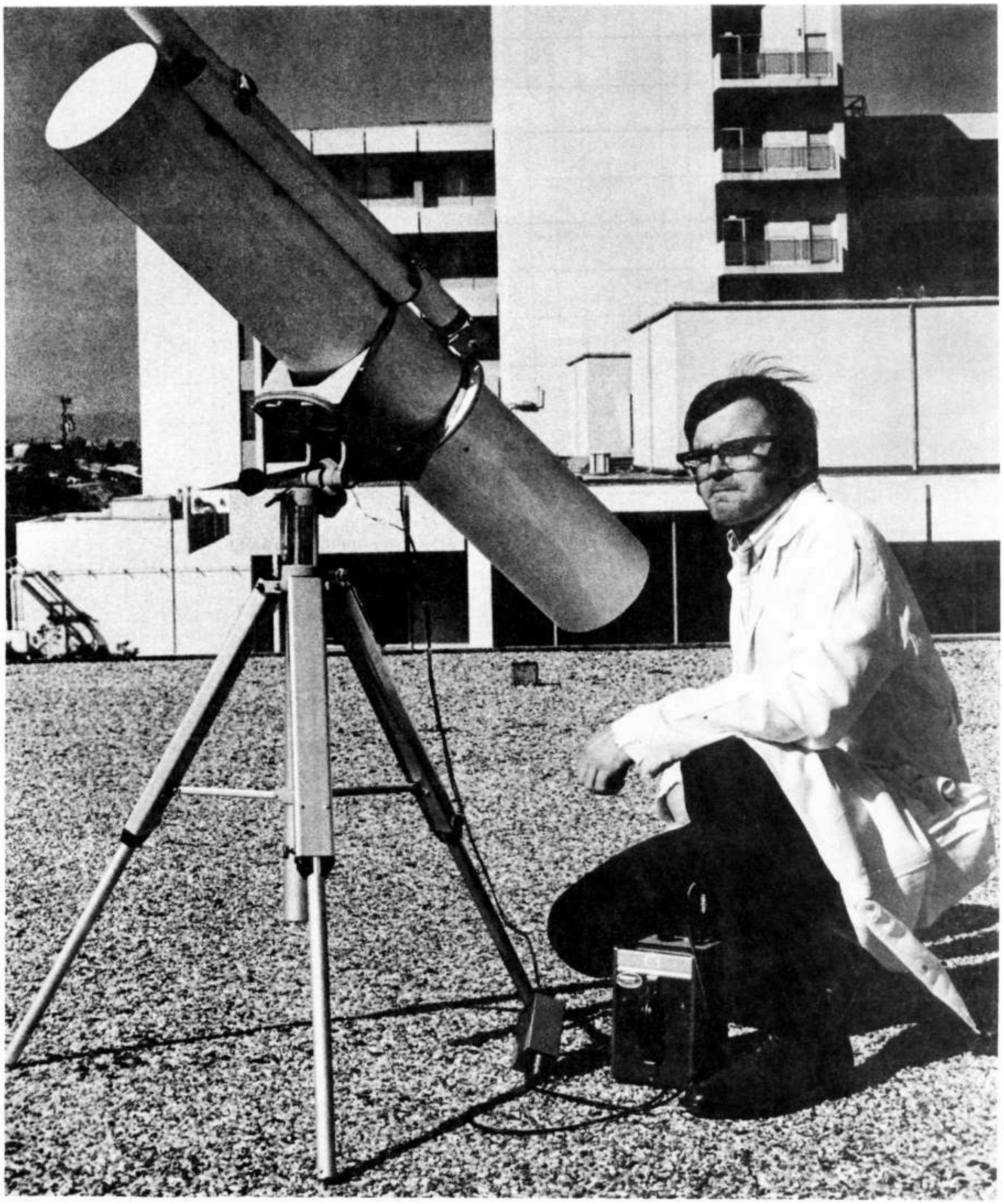

Fig. 4. Photograph of the solar aureole monitor

Ground Truth Obtained During ERTS-1 Passes

The bulk of our test area was covered by ERTS-1 sensors on Aug. 10, Aug. 28, Sept. 15. However; the passes on Aug. 9, Aug. 21, and Sept. 14 came sufficiently close for us to obtain reasonably good ground truth. We have attempted, and will continue to cover both locations, at least during the early phase of the program.

a) Airport visibility data

b) Air Pollution Control District Data

c) Aureole Monitor

d) Visual and Photographic data. Altitude of the top of the haze layer and vertical temperature profiles.

Table II. Notes on ERTS-1 Passes

Date	Coverage Obtained	Notes
Aug 9	b)	We were not yet informed of the ephemeris, so no active ground truth could be obtained.
Aug 10	b)	
Aug 27	a) b)	Cloudy
Aug 28	a) b)	Looked like it would be cloudy so no effort was launched to obtain active ground truth.
Sept 14	a) b) c) d)	Ideal conditions. Part of the area hazy, part clear, complete ground truth cloudy until minutes before pass. We couldn't get off the ground.

No images (70 mm pos or negs) were received for examination.

Accomplishments Planned for the Next Reporting Period

Continue to obtain ground truth at every opportunity. Examine the images from the 14 Sept pass and order digital tapes for analysis if the images are as expected.

@ABS The author has identified the following significant results.

A general purpose flexible computer program for the analysis of ERTS MSS data has been written for the Control Data 7600 computer. By putting all of the analysis operations into a separate subroutine the user may concentrate on the analysis. The problems of getting the information into the computer in convenient format is solved once and need not be tampered with each time a new analysis procedure is desired. The problems of getting information written by the XDS computers in 8 bit byte format into a convenient format for analysis on the 60 bit word format CDC 7600 computer are not trivial. A general purpose program to accomplish this has been written and is available from the investigators upon request.

No publications were prepared.

There are no recommended changes in operations and no changes contemplated or requested in standing order products.

No ERTS image description forms were submitted.

No data request forms for retrospection data were submitted.

REFERENCES

1. A.E.S. Green, A. Deepak, and B. J. Lipofsky, Appl. Opt., 10, 1263 (1971).

2. Reiner Eiden, Tellus, XX, 380 (1968).

3. Max Barn and Emil Wolf, Principles of Optics (Pergamon Press, 1959) pp 394-397.

APPENDIX

IBM Byte to CDC Word Conversion Program

This routine for use on Control Data 6000-7000 series computers convert information from records generated on 8 bit byte oriented computers, such as XDS computers or IBM 360-370 computers to convenient CDC formats. The routine is written in the CDC assembler language, COMPASS, and is compatible with CDC RUN Fortran.

Depending on the source of bytes for the conversion to CDC format the routine may be used in any of three modes.

Mode 1 Bytes are supplied by the user in the argument array.
Mode 2 Bytes are supplied by the user in a separate array.
*Mode 3 Bytes are supplied by the user as blocks on a file.

The different modes are indicated to the routine by the values in the single array which is the argument of the subroutine for all calls. The specific values will be discussed later.

The specific type of conversion desired is indicated by the entry point of the routine which is called. These entries are:

NXBYTE This entry returns bytes to the user as directed by the argument array. Bytes are right justified with zero fill. The bit pattern is exactly that appear ing in the desired bytes. No conversions are made.

IHEX The CEC display code corresponding to the hexidecimal equivalent of the bytes desired is returned right justified and blank filled.

*In mode 3 the file reading is done by the routine employing the Record Manager under the CDC SCOPE 2.0 operating system on the 7600 computer. Use of Mode 3 under different operating systems might require some modifications to the routine. Modes 1 and 2 are believed to be independent of operating system.

IEBCDC The CDC display code corresponding to the byte is returned right justified with blank fill. Upper and lower case letters are not distinguished. For the assignment of symbols in the CDC character set to the IBM hex codes and symbols see Table I. The symbol (display code 66_8) is returned for all bytes which do not correspond to a letter, a number, or one of the symbols defined in Table A-I.

INT2 The 1's complement integer corresponding to the 16 bit 2's complement integer provided in two bytes will be returned.

INT4 The 1's complement integer corresponding to the 32 bit 2's complement integer provided in four bytes will be returned.

REAL4 The CDC floating point number corresponding to the 32 bit IBM floating point number contained in four bytes will be obtained and returned.

REAL8 The single precision CDC floating point number obtained from the 64 bit IBM format floating point number contained in 8 bytes by rounding the 56 bit IBM fraction to 48 bits and converting to CDC format is returned.

SKPRCD No value is returned. This entry reads a new block from the file (Mode 3 operation) and initilizes the pointers so that the next call for bytes in sequence will start at the beginning of the new block.

Except for SKPRCD, which returns no values, all entry points are used as Fortran functions. All of the entries are called with one argument, an array of at most 10 words. (Mode 1 requires only a 2 word array.) The significance of the various words in the array is as follows:

Word 1 The byte location or bytes to be converted.

If word 2 is positive (mode 2 or 3) --

This word indicates the location of the next byte desired.

If word 1 is positive--

This word is the number of the first byte desired from the array or block for thi s conversion. The first byte of the array or block is numbered 1.

If word 1 is negative --

The next bytes in sequence will be obtained. On the first call the first byte of the block or array will be returned except in mode 2 operation a negative value is not allowed on the first call.

In both cases this word will be set to -1 on return to the user.

If word 2 is negative (mode 1)--

Word 1 contains the bytes to be converted to CDC format right justified with zero fill. The contents of word 1 will not be changed.

Word 2 The mode of operation and the number of bytes desired for conversion.

For NXBYTE this must be between 1 and 7, otherwise only one byt will be returned.

For IHEX if this word is positive the bytes are taken from the block or array being converted and the value of this word should be between 1 and 5. If it's value is 6 or 7, then 6 or 7 bytes will be extracted from the array or block but only the 6th or 7th byte will be converted. For values greater than 7 only one byte will be extracted and converted.

If the value of this word is between -1 and -5 the right hand number of bytes from word 1 of the argument array will be converted. If it is zero or less than -5 only the rightmost byte of word 1 of the argument array will be converted.

For REAL8 if this word is positive it's value will be ignored and it will be set to 7 on return.

If this word is negative the most significant byte of the 8 required (the IBM sign-exponent byte) will be taken as the righthand 8 bits of the complement of this word. The other 7 bytes will be taken from the righthand 56 bits of word 1 of the argument array.

For all other entries only the sign of this word is important. If the sign is positive the required bytes are drawn from the block or array being converted and word 2 will be set to the number of bytes required for that particular conversion.

If the sign of word 2 is negative the value of word 2 is ignored and the number of bytes required for the conversion are taken from word 1 of the argument array.

If bytes are being extracted from a block or array (word 2 positive, mode 2 or 3 operation) and if bytes beyond the end of the block or array are required by the combination of values appearing in words 1 and 2 than mode 3 operation is assumed and an attempt will be made to read a new block into the array.

The rest of the words in the argument array are needed only for mode 2 or 3 operation. See the discussion of the modes to determine which should be set by the user in each of the modes in operation.

Word 3 This word contains the length of the array or block in bytes.

Word 4 This word contains the number of bytes remaining in the array or block.

Word 5 This word contains the bit position in the word corresponding to the next byte to be obtained. This quantity is maintained by the routine and should not be modified externally.

Word 6 This word contains the address of the word in the array in which the next byte begins. This quantity is maintained by the routine and should not be modified externally.

Word 7 This word is not used at present.

Word 8 This word indicates the name of the file from which blocks are obtained for mode 3 operation. It may be an integer, n, from 1 to 99 to indicate the file name in the usual Fortran convention of TAPEn. It may also contain a file name in left justified zero filled display code. In either case the file name must be mentioned on the program card. No file card is required in the control cards. The file is defined by the routine to be RT=U, BT=K and RB=1 and so does not have the usual Fortran file definition.

Word 9 This word contains the address of the first word of the buffer array which contains the block or array to be converted in modes 2 and 3 operation. (This may be obtained conveniently with the Fortran function LOCF. See the examples.)

In mode 2 operation this buffer is filled by the user with the array he desires to make conversions from.

In mode 3 operation this buffer is filled from the file defined by word 8, one block at a time. The buffer must be long enough to contain the longest block expected. That length will be 8 times the maximum number of bytes expected divided by 60 and rounded to the next highest integer.

Word 10 This word contains a flat which indicates the status of the file after the most recent call to the routine. Possible values are:

0 No change has occurred in the file or buffer, or a new block has read in successfully from the file by SKPRCD.

1 Bytes beyond the end of the current array or block were requested. A new record was satisfactorily read in but no data has been extracted from it. The word and byte pointers are set so that the next call to the routine will start at the next call to the routine will start at the beginning of the block or array if word 1 of the argument array is negative at that time.

2 An end of section, end of partition or end of information was encountered on the read. The type of end of data will be indicated by a message in the OUTPUT file.

3 A parity error was encountered on the last read.

4 All other record manager detected errors on the last read are indicated by the value 4. The RM error number will be indicated in a error message put into the OUTPUT file.

The various modes of operation, which depend on the source of bytes for conversion, are indicated to the routine as described below.

Mode 1 The bytes to be converted are supplied by the user right justified in word 1 of the argument array. This mode is indicated by a negative sign in word 2 of the argument array. The value of word 2 is ignored except for two cases.

For REAL8 conversion word 2 is the complement of the most significant byte of the word.

For IHEX conversion word 2 is the complement of the number of bytes to be converted.

Mode 2 The bytes to be converted are extracted by the routine from an array supplied by the user. In this mode the following words are set, at least initially by the user.

Word 1 On the first call this word cannot be negative. A specific byte location must be specified. Subsequent calls may specify the byte desired explicitly or may obtain the bytes in sequence by setting (or leaving) this word set to a negative value.

Word 2 This word must be positive. See the word description for limitations on its value.

Word 3 This must be set by the user to the length, in bytes, of the array from which the bytes will be drawn. This must be done before the first call to the routine and need not be reset unless a new byte string of different length is put into the array by the user.

Word 4 This must be set by the user, before the first call to the routine, to the length, in bytes, of the array from which the bytes will be drawn, i.e., it is set to the same value as word 3. It shouldn'5 be reset by the user thereafter unless a byte string of different length is put into the array by the user.

Word 9 This word must contain the address of the first word of the array containing the bytes to be converted.

Words 5, 6, 7, 8 and 10 are not set by the user.

Mode 3 The bytes are extracted from blocks obtained from a file. In this mode the following words are set, at least initially by the user. In addition the file name must be mentioned in the main program card.

Word 1 This word may have any value allowed in the word description on any call.

Word 2 This word must be positive. See the word description for limitations on its value.

Word 4 This word should be set to a negative value on the first call of the program to assure reading the first block into the buffer. Thereafter this word need not concern the user.

Alternatively the first block may be input with the SKPRCD function and this word always ignored.

Word 8 This must indicate the file name from which the records will be obtained. See the word description for a discussion of possible forms.

Word 9 This word contains the address of the first word of a storage area used to store blocks as they come in from the file. The length required is discussed in the word description.

Words 3, 5, 6, 7, and 10 are not set by the user but may contain information of interest to him as described under the various word descriptions.

Length: The subprogram is 415_8 words long.

Other Routines Required: No other routines unavailable with a normal RUN Fortran job are called. Several RUN Fortran Library routines are called however in Mode 3 operation.

Error Messages: Error Messages are written on the OUTPUT file whenever a record manager error occurs or when an end of data is encountered in mode 3 operation.

Notes and Examples

1) IEBCDC can conveneiently be combined with the routine STRMOV to transform long strings of EBDCIC characters into display code character strings. For example we assume the mode of the subprogram will be 2. (User supplied arrays.) Assume that a user written function routine, IBYTES, is used to fill the array IAR with the bit pattern to be converted and that the function returns as its value the length in bytes of the array filled. Assume that in this particular array there is a strong of 108 EBDCIC characters starting with byte 325. It is desired to transform that string to display code, store it in an array ITST and print it out.

The following coding will fill the buffer array by means of the here undescribed routine IBYTES, set up all required words for mode 2 use, convert the string of characters desired and print it.

```
      COMMON IAR(512)
      DIMENSION ICONT(10),ITST(11),ITTM(2)
      ICONT(3) = IBYTES(IAR)
      ICONT(4) = ICONT(3)
      ICONT(9) = LOCF(IAR)
      ICONT(1) = 325
      ITST(11) = 1H
      DO 105 I=1,108
      CALL STRMOV(IERCDC(ICONT),10,1,ITST,I)
  105 CONTINUE
      WRITE (6,2) ITST
    2 FORMAT (* ITST = *,6A10,/8X,5A10)
```

The results are:

```
ITST = THIS ISA MESSAGE. NUMBERS - 01234567898765432l0 SYMBOLS ≥.<(
       +↓↑∨$*);≤-/,%⇏>∧:≡[]=≠  END OF CHARACTER STRING.
```

2) For calls to NXBYTE and IHEX the number of bytes to be converted may be, within limits, specified. The results of varying the number of bytes even beyond the allowed limits is indicated by the following coding, again applied to the character string of the preceeding example. This example also illustrates the use of a mode 1 call. This example presupposes the initialization coding of example 1.

```
      ICONT(1) = 325
      ITST(2) = -5
      DO 135 I = 1,9
      ICONT(2)=I-1
      ITST(1) = NXBYTE(ICONT)
      ITST(3) = IHEX(ITST)
      WRITE (6,5)  ITST(1),ITST(3),ITST(3)
    5 FORMAT (* ITST = *,020,5X,A10,5X,020)
  135 CONTINUE
      ICONT(1) = 325
      DO 140 I=1,6
      ICONT(2)=I
      IOUT = IHEX(ICONT)
      WRITE (6,5) IOUT,IOUT
  140 CONTINUE
```

The results of executing this are:

```
ITST = 00000000000000000343      00000000E3      33333333333333330536
ITST = 00000000000000000310      00000000C8      33333333333333330343
ITST = 00000000000000144742      000000C9E2      33333333333303440535
ITST = 00000000000020144742      000040C9E2      33333333373303440535
ITST = 00000000030120152305      00C140D4C5      33330334373304370340
ITST = 00000161342603437O5       E2E2C1C7C5      05350535033403420340
ITST = 00022640325711523O2       40D5E4D4C2      37330440053704370335
ITST = 03056636110030040360      E2406040F0      05353733413337330633
ITST = 00000000000000000361      00000000F1      33333333333333330634
ITST = 55555555555555550536              E3
ITST = 55555555555503430344            C8C9
ITST = 55555555053537330344           E240C9
ITST = 55550535033437330437         E2C140D4
ITST = 03400535053503340342       C5E2E2C1C7
ITST = 55555555555555550437              D4
```

3) These examples illustrates the us e of calls for number conversion. Assume the array of the previous examples has a sequence of 14 single IBM precisions (32 bit) floating point words starting at byte number 121. The following coding will extract these numbers and convert them to CDC floating point format. The other numberical conversions may be applied in exactly the same way, changing only the entry point name used to obtain the various conversions. This example presupposes the initialization coding of example 1.

```
      ICONT(1) = 121
      DO 110 I=1,14
      OUT = REAL4(ICONT)
      WRITE(6,3) OUT,OUT
    3 FORMAT (* OUT = *,E20.10,5X,O20)
  110 CONTINUE
```

The results of executing this are:

```
 OUT =      .1000000000E+01    17204000000000000000
 OUT =     -.1000000000E+01    60573777777777777777
 OUT =      .1492500000E+03    17274524000000000000
 OUT =     -.1492500000E+03    60503253777777777777
 OUT =     0.                  00000000000000000000
 OUT =     -.5000000000E+00    60603777777777777777
 OUT =      .1562500000E-01    17124000000000000000
 OUT =     -.1562500000E-01    60653777777777777777
 OUT =     -.1500000000E+02    60540377777777777777
 OUT =      .1500000000E+02    17237400000000000000
 OUT =      .5397605347E-78    13144000000000000000
 OUT =     -.5397605347E-78    64633777777777777777
 OUT =      .7237005146E+76    23147777777700000000
 OUT =     -.7237005146E+76    54630000000077777777
```

Table A-1

IEBCDC Symbol Conversion Code

HEX Value	IBM Symbol	CDC Display Code (Octal)	CDC Symbol
40	space	55	blank
4A	¢	75	≥
4B	.	57	.
4C	<	72	<
4D	(	51	(
4E	+	45	+
4F	|	71	↓
50	&	70	↑
5A	!	66	∨
5B	$	53	$
5C	*	47	*
5D	)	52	)
5E	;	77	;
5F	¬	74	≤
60	-	46	-
61	/	50	/
6B	,	56	,
6C	%	76*	¬*
6D	—	65	↱
6E	>	73	>
6F	?	67	∧
7A	:	63	:
7B	#	60	≡
7C	@	61	⊏
7D	'	62	⊐
7E	=	54	=
7F	"	64	≠

*Octal 76 prints % on 200 terminals and ¬ on high speed Line printers.